

Prediction Performance of End-Milling Process by Gene Expression Programming

Mohammed T. Hayajneh^{*a}, Majid Abdellahia^b

^aIndustrial Engineering Department, Faculty of Engineering, Jordan University of Science and Technology

^bAdvanced Materials Research Center, Islamic Azad University, Najafabad Branch, Najafabad, Iran

Received 31 March, 2019

Abstract

In this study, two different gene expression programming models were applied to predict surface roughness of end milling. The differences between the two models were the number of genes, chromosomes, head size, and the linking function. To construct the models, 84 pair input-target data were collected by the experimental procedure, randomly parted into 60 and 24 data sets and then were trained and tested respectively by the suggested models. The spindle speed, cutting feed and depth of cut were the independent input parameters. According to these input parameters, the roughness of the surface in the end-milling process at different cutting conditions was predicted. The training and testing results in the gene expression programming models have presented an acceptable potential for predicting roughness values of end-milling in the considered range.

© 2019 Jordan Journal of Mechanical and Industrial Engineering. All rights reserved

Keywords: gene expression programming; end-milling; modeling; surface roughness;

1. Introduction

Surface roughness is one of the most common element measurements in the machining processes. Surface roughness is tangible parameter to quantify the quality of the machined surfaces. In machining processes, surface roughness is needed to be as low as possible. Modeling techniques for the prediction of surface roughness (Ra) can be categorized into three groups which are analytical models, experimental models and Artificial Intelligence (AI)-based models [1]. Analytical and experimental models can be developed by using predictable approaches such as the statistical regression technique. On the other hand, AI-based models are established using non-conventional methodologies such as Fuzzy Logic, Artificial Neural Network, Genetic programming, and gene expression programming (GP) [1].

Different fuzzy logic and artificial neural network schemes have been broadly used for the selection of the working conditions in machining processes [2-10]. Gene expression programming gained broad consideration due to its capability to model nonlinear relationships for input-output mappings. Several studies have employed gene expression programming models for building industry problems. Aldas et al. [11] developed a genetic operation tree to study the effect of machining parameters and reinforcement content on thrust force during drilling of hybrid composites. Sener and Kurtarn [12] employed a genetic algorithm to optimize process parameters for

rectangular cup deep drawing. Yeh and Lien [13] developed a genetic operation tree to predict concrete strength. Vijaykumar et al. [14] applied gene expression programming to control the parameter of bidirectional CFRP composite pip.

In the present study, surface roughness as a performance indicator of end milling at different variations of spindle speed (N), cutting feed (F) and depth of cut (D) has been modeled by Gene expression programming. A total number of 84 data were collected from the experimental procedures, trained and tested using gene expression programming. The obtained results were compared by experimental ones to test the power of genetic programming for forecasting the surface roughness in the end milling process.

2. Experimental Procedure

The experiment used a Bridgeport end-milling machine. Eight 3/4" four-flute HSS cutting tools were used. Dry machining has been employed. The experiment was performed on aluminum work pieces [2]. Figure 1 shows the experiment setup. The cutting parameters were set as: four levels of spindle speed (750, 1000, 1250, 1500 rpm), seven levels of feed rate (150, 225, 300, 375, 450, 525, 600 mm/min), and three levels of depth of cut (0.25, 0.75, 1.25 mm). In this experimental study, the roughness measurements for surfaces were repeated three times using micro-meters. The measured surface roughness was the response variable. The surface roughness data were

* Corresponding author e-mail: hayajneh@just.edu.jo.

collected randomly for each of the 84 machining conditions defined by the levels of independent variables. Among 84 collected experimental sets, 60 sets were randomly chosen as a training set for the genetic programming models (Table 1) and the remaining 24 sets were used as testing the generalization capability of the proposed models (Table 2).

3. Genetic Programming and Gene Expression Programming Theory

Genetic programming (GP) is proposed by Koza [15]. It is a generalization of genetic algorithms (GAs) [16]. Genetic programming attempts to use computer programs as its data representation. Similarly, to GA, GP needs only the problem to be defined. Then, the program searches for a solution in a problem-independent manner [16]. Genetic programming breeds computer programs to solve problems by implementing the following three steps:

1. Create an initial population of random compositions of the functions and terminals of the problem.
2. Execute iteratively the following sub steps until the termination criterion has been satisfied:
 - Execute each program in the population and allocate the fitness value using the fitness measure.
 - Create a new population of computer programs by applying the following operations: Reproduction: (i) Copy an existing program to the new population, (ii) Crossover: Generate new offspring program(s) for the new population by recombining arbitrarily chosen parts of two existing programs and (iii) Mutation. Create one new offspring program for the new population by arbitrarily changing a randomly chosen part of one existing program.
3. The program that is categorized by the method of result designation is selected as the result of the genetic programming system for the run. This result may be a solution (or approximate solution) to the problem [16]. A flowchart of a typical Genetic programming algorithm is revealed in Figure 2 [17]. The genetic programming approach progresses through the action of three basic genetic operators: reproduction, crossover, and mutation. In the reproduction stage, an approach must be implemented as to which programs should die. In the implementation, a small proportion of the trees with the worst fitness are killed. The population is then filled with the surviving trees according to accepted selection mechanisms, as explained by Sarıdemir [17]. Crossover swamps randomly selected parts of two trees to join good information from the parents and to develop the fitness of the next generation, as shown in Figure 3 [16]. Mutation protects the model against premature convergence and develops the non-local properties of the search, as shown in Figure 4 [16]. Sometimes, one randomly selected node is interchanged by another one from the same set, except itself.

In applying genetic programming with automatic function definition to solving a problem, five major preparatory steps are used. These steps involve determining terminal set, function set, fitness function, control parameter and termination criteria [18].

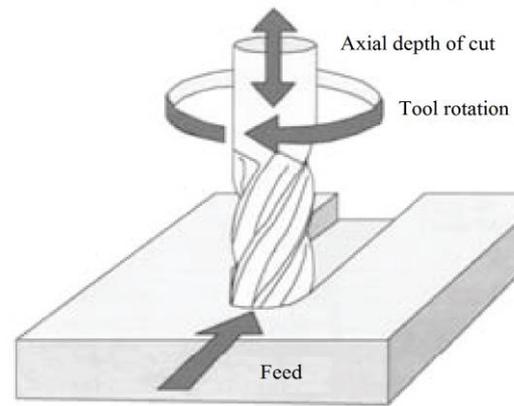


Figure 1. A schematic representation of an end-milling system [2].

Table1. Effect of machining parameters on the surface roughness (training set data set) [2].

No.	N RPM (rpm)	F (mm/min)	D (mm)	Ra (μm)	No.	N (rpm)	F (mm/min)	D (mm)	Ra (μm)
1	750	525	1.25	3.7	31	1000	600	0.25	4.1
2	1250	300	1.25	2.4	32	1500	150	0.25	1.3
3	1000	375	0.25	2.6	33	750	375	1.25	2.6
4	750	600	1.25	4.4	34	1500	525	1.25	3
5	750	300	0.75	2.6	35	1250	300	0.25	2.6
6	1500	375	1.25	2.5	36	1000	300	0.25	3.1
7	1250	450	1.25	2.3	37	1500	225	0.25	1.4
8	1000	300	1.25	2.3	38	750	225	0.75	2.6
9	750	150	1.25	1.9	39	750	150	0.75	1.7
10	1500	600	0.75	2.6	40	750	525	0.75	4
11	1500	450	0.25	3.2	41	1000	600	0.75	4
12	1000	450	0.25	4	42	1250	150	1.25	1.7
13	750	375	0.75	3.1	43	1000	375	0.75	2.6
14	1250	600	0.25	3.8	44	1250	300	0.75	2.5
15	1250	225	0.75	2.1	45	1000	225	0.75	2.4
16	1000	150	1.25	1.6	46	1500	300	0.75	2.1
17	1000	300	0.75	2.1	47	1000	525	0.75	3.9
18	750	450	1.25	3.3	48	1250	225	0.25	2.1
19	1500	600	0.25	3.2	49	1000	150	0.75	1.9
20	1250	525	0.75	2.5	50	1250	375	0.75	2.5
21	1500	450	1.25	2.6	51	1000	150	0.25	1.6
22	750	600	0.75	4.5	52	1000	225	1.25	2.7
23	1000	525	0.25	3.8	53	750	225	1.25	2.5
24	750	300	1.25	2.4	54	1250	450	0.75	2.2
25	1500	225	0.75	1.9	55	1500	300	0.25	2.3
26	1250	150	0.25	1.2	56	750	450	0.25	4.8
27	1250	525	1.25	2.5	57	1250	600	0.75	2.6
28	1250	375	1.25	2.5	58	750	525	0.25	4.5
29	1000	225	0.25	2.3	59	1250	225	1.25	2.4
30	1000	450	0.75	3	60	1250	150	0.75	1.7

Table 2. Effect of machining parameters on the surface roughness (testing set data set) [2].

No.	N (rpm)	F (mm/min)	D (mm)	Ra (μm)	No.	N (rpm)	F (mm/min)	D (mm)	Ra (μm)
1	1000	450	1.25	2.1	13	1250	600	1.25	3.1
2	1500	150	1.25	1.5	14	1250	375	0.25	2.7
3	750	300	0.25	3	15	1000	600	1.25	2.1
4	750	450	0.75	3.7	16	1500	300	1.25	2.4
5	1000	375	1.25	2.6	17	1500	525	0.75	2.6
6	750	225	0.25	2.1	18	1500	525	0.25	3.1
7	1500	375	0.25	2.7	19	1500	600	1.25	3.2
8	1250	525	0.25	3.1	20	750	150	0.25	1.6
9	1500	450	0.75	2.3	21	1250	450	0.25	2.5
10	750	600	0.25	4.7	22	1500	150	0.75	1.4
11	1500	375	0.75	2.1	23	1000	525	1.25	1.5
12	750	375	0.25	3.2	24	1500	225	1.25	1.8

Genetic programming (GP) has two principal elements such as the chromosomes and the expression trees (ETs). The chromosomes may be involved in one or more genes which refer to a mathematical expression. The mathematical code of a gene is identified in two different languages called Karva Language [16]; such as the language of the genes and the language of the expression trees (ETs). The genes have two main parts addressed as the head and the tail. The head comprises some mathematical operators, variables and constants (+, -, *, /, $\sqrt{\quad}$, sin, cos, 1, a, b, c) which are used to encrypt a mathematical expression. The tail just comprises variables and constants (1, a, b, c) named as terminal symbols. Additional symbols are used if the terminal symbols in the

head are insufficient to define a mathematical expression. A simple chromosome as a linear string with one gene is encrypted in Figure 5. Its ET and the corresponding mathematical equation are also shown in the same figure. The translation of ET to Karva Language is done by starting to read from left to right in the top line of the tree and from top to bottom. The arrangements of genes used in this method are like the arrangements of biological genes and have coding and non-coding parts. On the other hand, more complex mathematical equations are defined by more than one chromosome denoted to multigenic chromosomes. Joining the genes is done by combining functions such as addition, subtraction, multiplication, or division [18].

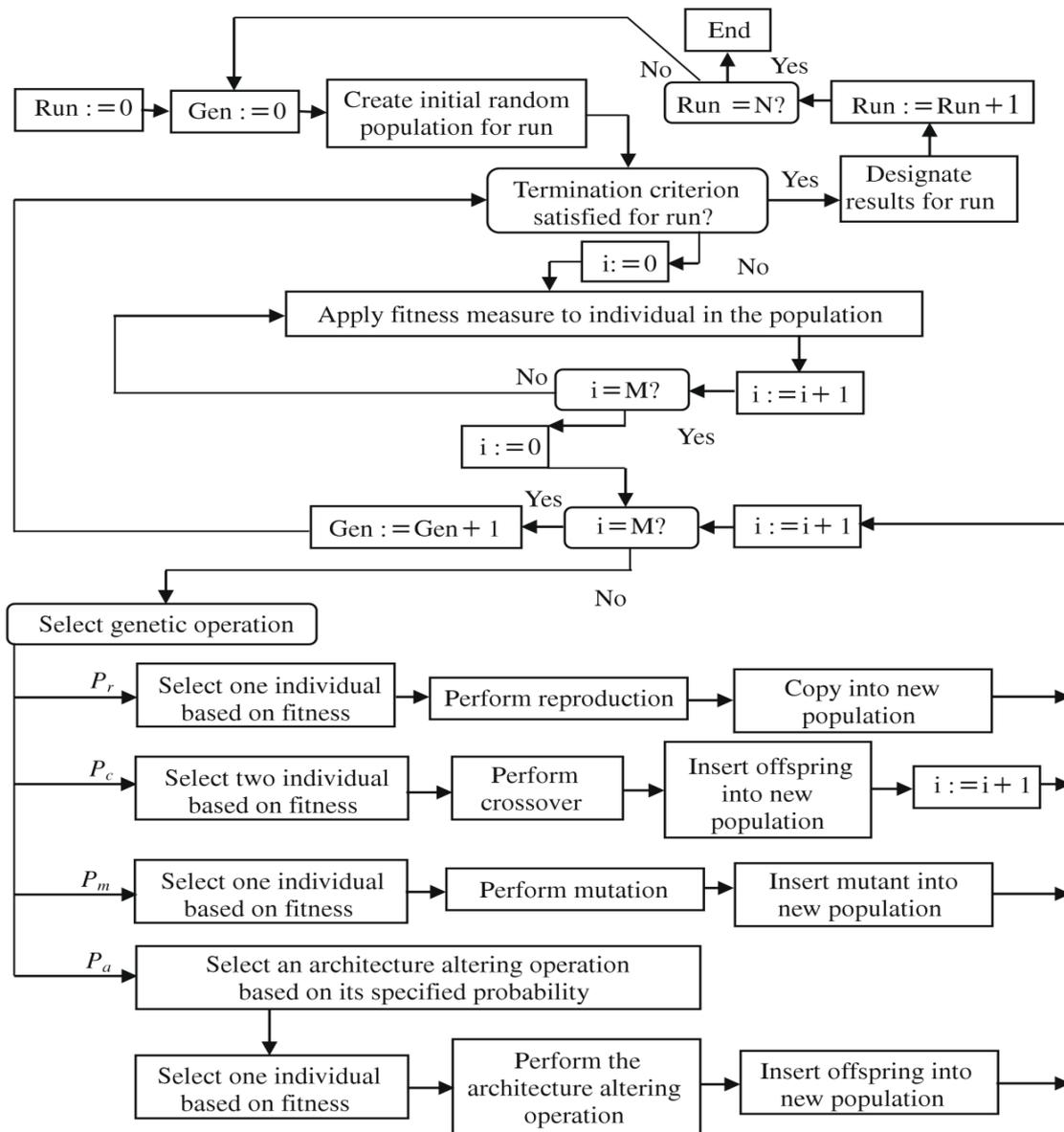


Figure 2. Genetic programming flowchart.

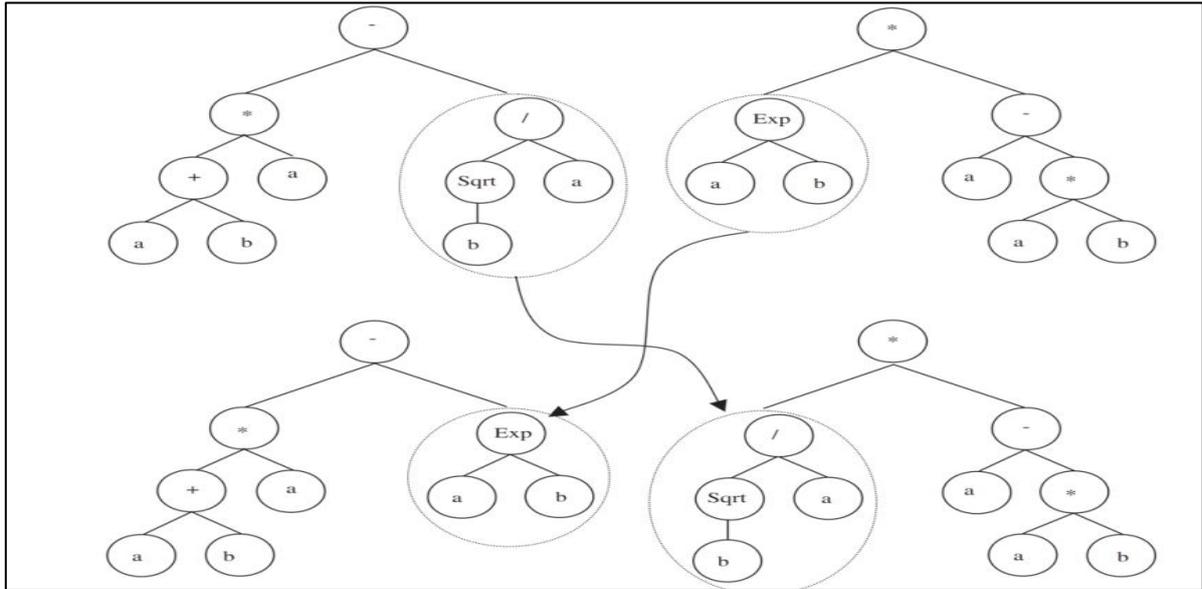


Figure 3. Example of genetic programming crossover

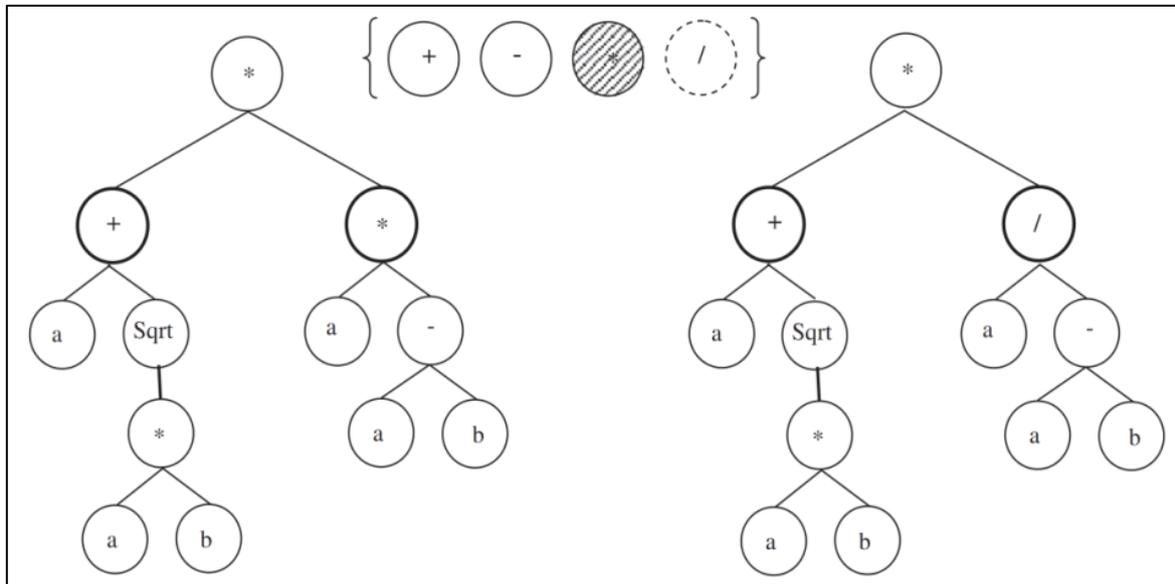


Figure 4. Example of genetic programming mutation

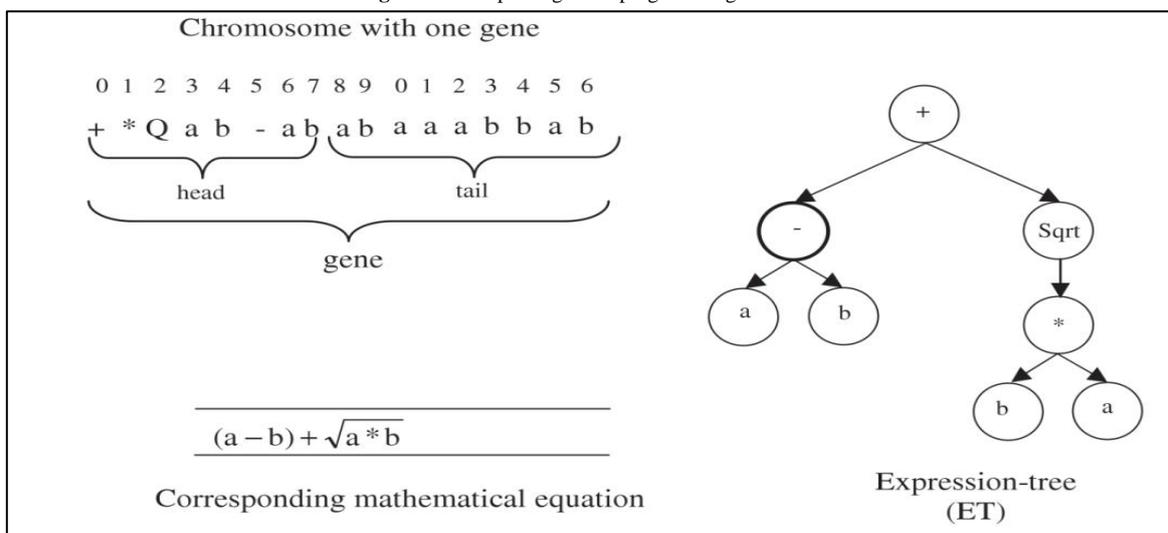


Figure 5. Chromosome with one gene and its expression tree and corresponding mathematical equation

4. Gene Expression Programming Structure and Parameters

In this study the expression trees of two different genetic programming models which were termed GP-I and GP-II were constructed for Ra values of the end-milling process. In the GP-I and GP-II, the number of genes was six and seven (Sub-ETs), and linking function was addition and subtraction, respectively. In the training and testing of GP-I and GP-II models, N, F and D were considered as input data and Ra as independent output data. Among 84 experimental sets, 60 sets were arbitrarily selected as a training set for the GP-I and GP-II modeling and the remaining 24 sets were used as testing the generalization capacity of the proposed models.

For the genetic programming-based formulations, the first is to select the fitness function. For this problem, firstly, the fitness, f_i , of an individual program, i , is measured by Eq. (1):

$$f_i = \sum_{j=1}^{C_t} (M - |C_{(i,j)} - T_j|) \quad (1)$$

where M is the range of selection, $C_{(i,j)}$ is the value returned by the individual chromosome i for fitness case j (out of C_t fitness cases) and T_j is the target value for fitness case j . If the precision $|C_{(i,j)} - T_j|$ is less than or equal to 0.01, then the precision is equal to zero, and $f_i = f_{max} = C_t M$. In this case, $M = 100$ was used, therefore, $f_{max} = 1000$. The benefit of this type of fitness functions is that the systems can find the best solution for itself. The second significant step makes up by selecting the set of terminals T and the set of functions F to generate the chromosomes. In this problem, the terminal set comprises clearly of the independent variable, i.e., $T = \{N, F \text{ and } D\}$. The select of the proper function set is not so obvious, but an appropriate guess can always be done to include all the required functions. In this situation, four basic arithmetic operators (+, -, *, /) and some basic functions (Sqrt, x^2 , x^3 , ln, sin, cos, Arctan, Exp) [19].

The third important step is to determine chromosomal tree, i.e., head length and number of genes. Genetic programming-based formulations firstly use single gene and lengths of 2 heads, and increase the number of genes and heads, one by one while running, and examined the training and testing performance for each formulation [19]. In this study, after several trials, for all the genetic programming-based formulations, number of genes and the length of the head designed as given in table 3. The fourth major step is to choose the linking function. For GP-I and GP-II models, addition and subtraction were created for linking functions, respectively.

Finally, a combination of all genetic operators (mutation, transposition and crossover) was employed as

$$R_{a(GP-I)} = \frac{(\sin(-1.36) + \ln N) + (N^3 \times \ln 2.88)}{\sqrt{F}} + \frac{\cos(D+F-8.93)}{(\tan^{-1} 8.93 - \cos D)(-6.05N)} + \frac{\sqrt{F}}{\ln((N \tan^{-1} 0.34)(D-F) - 9.92)} + \frac{\cos(((N+F) - 584.27)(N \cos -7.17))}{\ln D} + \frac{\cos((F+4.42)(1.26N))}{(\sqrt{4.42-4.42N})(3.50)} - 1.68 + \tan^{-1}(\sqrt{148.03 \tan^{-1} N} + \sin F^3) \quad (6)$$

$$R_{a(GP-II)} = \frac{((9.92-F)N)^2}{2(D-F)} - \sqrt[3]{N} - (\sqrt[3]{2D-9.74})(\ln F - \sqrt[3]{F}) - \exp^{\frac{1}{3}}(\sin N + \cos(93.89N)) - \cos^3 F - \frac{(\cos(7.35+D))(F-9.96)}{(D-9.96)\sqrt[3]{N}} - \tan^{-1} \left(\left(((9.71-F) - 841.23) + (F/N) \right) - D \right) \quad (7)$$

set of genetic operators. At first, these parameters were well thought-out as the program defined values. Then they were increased step by step and the utmost performance network based on the R^2 values was considered. Parameters of the training of the GP-I and GP-II approach models are given in Table 4. For the GP-I and GP-II approach models, chromosome 32 and 42 were observed to be the best of generation individuals forecasting surface roughness. Explicit formulations based on the GP-I and GP-II approach models for R_a were attained by:

$$R_a = f(N, F, D) \quad (2)$$

The related formulations could be found by the procedure shown in Figure 5.

Table 3. Parameters of GP approach models

Parameter definition	GP-I	GP-II
Chromosomes	32	42
Head size	8	10
Number of genes	6	7
Linking function	addition	subtraction
Mutation rate	0.044	0.044
Inversion rate	0.1	0.1
One-point recombination rate	0.3	0.3
Two-point recombination rate	0.3	0.3
Gene recombination rate	0.1	0.1
Gene transposition rate 0.1	0.1	0.1
Constants per gene	5	5
Weight of functions	7	7
Lower bound	10	10
Upper bound	10	10

5. Results and Discussion

In the present study, in order to assess the capabilities of genetic programming-based formulations, formulas given by some national building codes and the established regression-based formulation, R-square (R^2), root-mean-squared error (RMSE) and mean absolute percentage error (MAPE) were used as the criteria between the experimental and predicted values which are according to the equations (3) – (5), respectively [19]:

$$R^2 = \frac{(n \sum t_i o_i - \sum t_i \sum o_i)^2}{(n \sum t_i^2 - (\sum t_i)^2)(n \sum o_i^2 - (\sum o_i)^2)} \quad (3)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (t_i - o_i)^2} \quad (4)$$

$$MAPE = \frac{1}{n} \left[\frac{\sum_{i=1}^n |t_i - o_i|}{\sum_{i=1}^n t_i} \times 100 \right] \quad (5)$$

Where t is the experimental result, o is the obtained result by the formulations and n is the total of data. The related equations to GP-I and GP-II models are in accordance to equations (6) and (7), respectively:

The experimental and the predicted are shown in Figure 6. R-square (R^2), root-mean-squared error (RMSE) and mean absolute percentage error (MAPE) values were shown in Table 4 for the training and testing data. As shown in Figure 6, the results attained from the training and testing in GP-I and GP-II models are in good agreement with the experimental results. As seen in Figure 6a, b and c, d the predicted results from models are compared to the experimental results for training, testing and validation sets, respectively. The training set results demonstrated that the proposed models have remarkably well learned the non-linear relationship between the input and the output variables with high correlation and reasonably low error values. Comparing the GP-I and GP-II approach models' prediction with the experimental results for the testing and training stages proves a high generality capacity of the proposed models and low error values. All these findings show a successful performance of the models for predicting surface roughness in end-milling in training and testing stages. The result of testing phase in Figure 6 shows that the GP-I and GP-II models are capable of generalizing between input and output variables with reasonably good predictions. The performance of the GP-I and GP-II models is shown in Table 4. The best value of R^2 and the minimum value of MAPE and RMSE are 0.923, 0.268 and 0.219, respectively, all for training set in GP-I model. The minimum value of R^2 and the maximum value of MAPE and RMSE are 0.901, 0.339 and 0.471, respectively, all for testing set in GP-II model. The entire R^2 , MAPE and RMSE values show that the proposed GP-I and GP-II models are appropriate and can predict surface roughness

values of end-milling very close to the experimental values.

Table 4. Statistical calculations from GP-I and GP-II training and testing phases

Statistics	GP-I		GP-II	
	Training	Testing	Training	Testing
R^2	0.923	0.920	0.909	0.901
MAPE	0.268	0.314	0.296	0.339
RMSE	0.219	0.411	0.249	0.471

6. Conclusions

This study reports original and efficient models for the formulation of surface roughness in the end-milling process. Two different GP-I and GP-II models were proposed to forecast the roughness values in the end-milling process. The suggested models were based on experimental results. The number of genes in the proposed GP-I and GP-II models was 6 and 7, and the linking functions were addition and subtraction, respectively. All the results attained from the models exhibit excellent agreement with experimental results. The best value of R^2 and the minimum value of MAPE and RMSE are 0.923, 0.268 and 0.219, respectively, all for training set in GP-I model. The minimum value of R^2 and the maximum value of MAPE and RMSE are 0.901, 0.339 and 0.471, respectively, all for the testing set in GP-II model. It was found that GP can be a substitute approach for the assessment of the performance of end-milling process and can calculate their surface-roughness value with suitable input parameters.

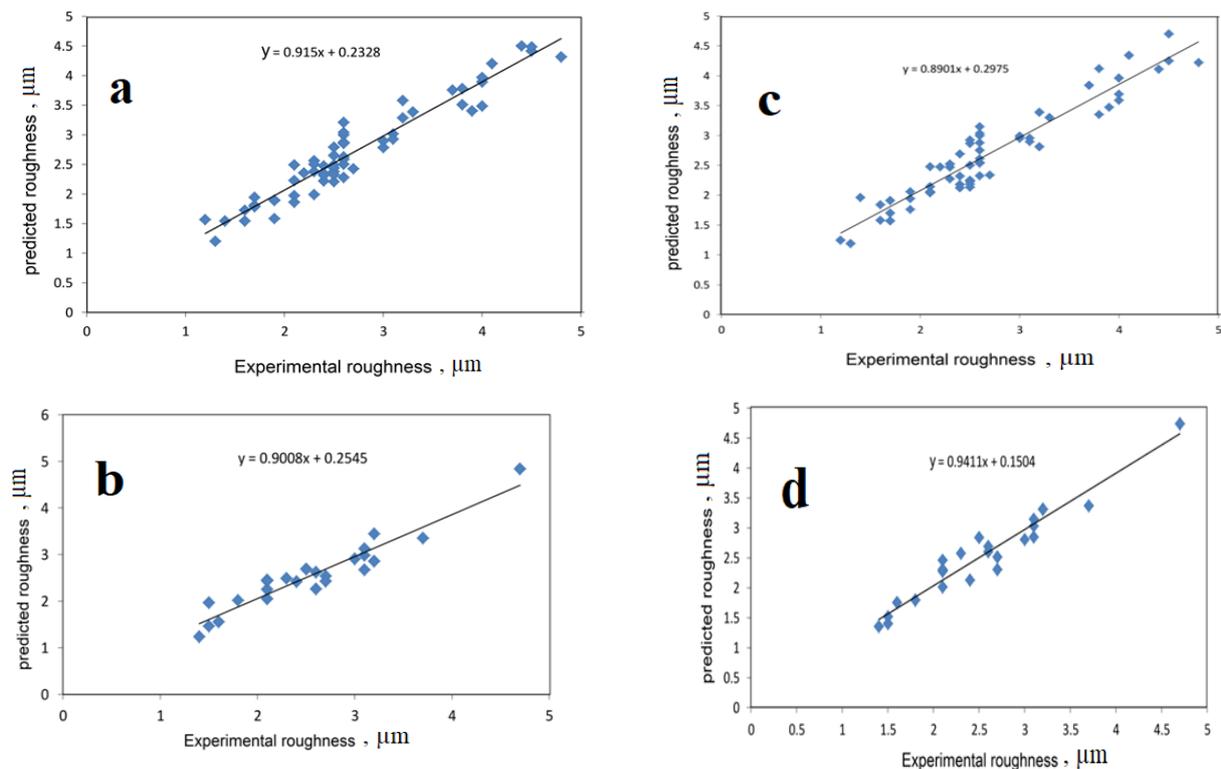


Figure 6. The results obtained from experimental studies and predicted (Model) by using the training and testing results of GP-I and GP-II models. Respectively, (a), (b) training and testing related to GP-I model; (c), (d) training and testing related to GP-II model

References

- [1] P. Benardos and G. osnaikos, Predicting surface roughness in machining: a review. *International Journal of Machine Tools and Manufacture*, vol. 43, No.8, pp. 833–844, 2003.
- [2] M. Hayajneh, M. Tahat, J. Bluhm, A Study of the Effects of Machining Parameters on the Surface Roughness in the End-Milling Process. *Jordan Journal of Mechanical and Industrial Engineering*, vol. 1, no. 1, pp. 1-5, 2007.
- [3] M. Hayajneh, S. Radaideh, Modeling surface finish in end milling process using fuzzy subtractive clustering based system identification method, *Materials and Manufacturing Processes*, vol. 18, no. 4, pp. 653-665, 2003.
- [4] M. Hayajneh, Fuzzy clustering model for surface finish prediction in fine turning process, *International Journal of Machining Science and Technology*, vol. 9, no. 3, pp. 437-451, 2005.
- [5] M. Hayajneh, S. Radaideh, Monitoring hole quality in drilling process using fuzzy subtractive clustering based system identification method, *Journal of Testing and Evaluation*, vol. 35, no. 3, pp. 297-302, 2007.
- [6] M. Hayajneh, A. Hassan, Modelling the machinability of self-lubricated aluminum/alumina/graphite hybrid composites using a fuzzy subtractive clustering-based system identification method, *International Journal of Machining and Machinability of Materials*, vol. 3, no. 3/4, pp. 252-271, 2008.
- [7] H. Al-Wedyan, M. Hayajneh, Prediction and controlling of roundness during the BTA deep hole drilling process: Experimental investigations and fuzzy modeling, *Materials Testing*, vol. 59, no. 3, 2017.
- [8] Hassan, A. Alrashdan, M. Hayajneh, A. Mayyas, Prediction of density, porosity and hardness in aluminum–copper-based composite materials using artificial neural network, *Journal of Materials Processing Technology*, vol. 209, no. 2, pp. 894-899, 2009.
- [9] M. Hayajneh, A. Hassan, A. Alrashdan, A. Mayyas: Prediction of tribological behavior of aluminum–copper based composite using artificial neural network, *Journal of Alloys and Compounds*, vol. 470, no.1, pp. 584-588, 2009.
- [10] M. Hayajneh, A. Hassan, A. Mayyas, Artificial neural network modeling of the drilling process of self-lubricated aluminum/alumina/graphite hybrid composites synthesized by powder metallurgy technique, *Journal of Alloys and Compounds*, vol. 478, no.1, pp. 559-565, 2009.
- [11] K. Aldaş , I. Özkul and M. Hayajneh, Effects of machining parameters and reinforcement content on thrust force during drilling of hybrid composites, *Materials Testing*, vol. 58, no. 3, pp.280-284, 2016.
- [12] Sener and H. Kurtaran, Optimization of process parameters for rectangular cup deep drawing by the Taguchi method and genetic algorithm, *Materials Testing*, vol. 58, no.3, pp. 238-245, 2016.
- [13] Yeh, L. Lien L, Knowledge discovery of concrete material using Genetic Operation Trees. *Expert Systems with Applications*, vol. 36, no.3, pp. 5807-5812, 2009.
- [14] K. Vijaykumar, K. Panneerselvam, and A. Naveen Sait, Machining parameter optimization of bidirectional cfrp composite pipe by genetic algorithm, *Materials Testing*, vol. 56, no.9, pp. 728-736, 2014.
- [15] J. Koza, Genetic programming as a means for programming computers by natural selection. *Statistics and Computing*, vol. 4, no.2, pp. 87-112, 1994.
- [16] Kara, Prediction of shear strength of FRP-reinforced concrete beams without stirrups based on genetic programming. *Advances in Engineering Software*, vol. 42, no.6, pp. 295-304, 2011.
- [17] M. Saridemir, Genetic programming approach for prediction of compressive strength of concretes containing rice husk ash, *Construction and Building Materials*, vol. 24, no.10, pp. 1911-1919, 2010.
- [18] M. Khan, A. Alam, Survey of application, genomics and genetic programming, a new frontier, *Genomics*, vol. 100, no.2, pp. 65-71, 2012.
- [19] M. Saridemir, Empirical modeling of splitting tensile strength from cylinder compressive strength of concrete by genetic programming, *Expert Systems with Applications*, vol. 38, no. 11, pp. 14257-14268, 2011.